Refine Search

Search Results -

Terms	Documents
L7 same path	15

Database: US Pre-Grant Publication Full-Text Database; US Patents Full-Text Database; US OCR Full-Text Database; EPO Abstracts Database; JPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Search: L23

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Thursday, February 16, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L23	L7 same path	15	L23
L22	L7 and l2	15	L22
L21	L20 and l7	1	L21
L20	L2[ti]	20	L20
L19	L18 and l17	13	L19
L18	L2 same pin	38	L18
L17	L16 and l15	15	L17
L16	scan adj1 command	658	L16
L15	L14 and l7	15	L15
L14	L13 and l2	16	L14
L13	L12 and emulating	16	L13
L12	L11 and scan	23	L12
L11	L2 and jtag	25	L11
L10	L9 same emulating	1	L10

L9	L1 same jtag	102	L9
L8	L7 same l1	2	L8
L7	emulating same scan	140	L7
L6	L2 same emulating	5	L6
L5	L3 and l1	1	L5
L4	L3 same emulating	0	L4
L3	virtual adj1 scan adj1 path	9	L3
L2	embedded adj1 l1	510	L2
L1	microcontroller	38134	L1

END OF SEARCH HISTORY

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L6: Entry 3 of 5 File: USPT Jul 19, 1994

DOCUMENT-IDENTIFIER: US 5331571 A
TITLE: Testing and emulation of integrated circuits

Abstract Text (1):
An architecture is provided for testing and emulating an integrated circuit with embedded function blocks. The output nodes of the function blocks are connected through a tri-state buffer to a test bus which in turn is connected to configurable external pins. The external pins multiplex the normal I/O in normal mode and the test bus I/O in the test mode. The test bus is also connected through multiplexers to input nodes of function blocks. In test mode, the function block nodes are accessed through the test bus. For emulation of an embedded microcontroller or microprocessor, the internal connections of the microcontroller (or microprocessor) are brought out to those external pins which in normal operation are connected only to the microcontroller and not to any other function block. An in-circuit emulator (ICE) emulating the microcontroller is connected to the other function blocks through those external pins.

Detailed Description Text (64):
Thus microcontroller 120.1 in the integrated circuit 2110 of FIG. 22 can be emulated by the same emulator 3030 that is used for a discrete, non-embedded microcontroller. The need for an emulator emulating the whole integrated circuit 2110 is eliminated. The need for discrete devices to replace function blocks 120.2, 120.3, 120.4 is also eliminated. The same or identical function blocks 120.2, 120.3, 120.4 are used in the normal and the emulation modes. High degree of precision during emulation is thereby provided because this architecture allows a good emulation of the normal-mode operation of circuit 2110.

Previous Doc Next Doc Go to Doc#

First Hit Previous Doc Next Doc Go to Doc#

End of Result Set

Generate Collection Print

L10: Entry 1 of 1 File: JPAB Jul 4, 2003

PUB-NO: JP02003186693A
DOCUMENT-IDENTIFIER: JP 2003186693 A
TITLE: MICROCONTROLLER HAVING EMULATING FUNCTION

PUBN-DATE: July 4, 2003

INVENTOR-INFORMATION:

NAME	COUNTRY
SAKAMOTO, KEI	
NASU, TAKASHI	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
MITSUBISHI ELECTRIC CORP	

APPL-NO: JP2001384107
APPL-DATE: December 18, 2001

INT-CL (IPC): G06 F 11/22; G06 F 11/28; G06 F 15/78

ABSTRACT:

PROBLEM TO BE SOLVED: To prevent leakage of a resource, when it is being debugged.

SOLUTION: The debugging function control circuit 1500 of a microcontroller, having an emulating function includes a debug IF input/output circuit 1560 connected with a JTAG-ICE (In-circuit Emulator), a function restriction mode register 1584, an access key register 1582, a protection checking circuit 1580 and a resource access circuit 1520. When a debugging command is inputted from the JTAG-ICE, the protection checking circuit 1580 transmits a debugging command thus inputted to the resource access circuit 1520, when an operation mode stored in the function restriction mode register 1584 is not a restrictive mode. The protection checking circuit 1580 rewrites the operation mode for a permission mode, if the access key inputted from the JTAG-ICE match with the access key stored in the access key register 1582.

Previous Doc Next Doc Go to Doc#

L12: Entry 2 of 23 File: USPT Oct 11, 2005

DOCUMENT-IDENTIFIER: US 6954879 B1
TITLE: Method and apparatus for communicating configuration data for a peripheral device of a microcontroller via a scan path

Abstract Text (1):
A microcontroller has many internal peripheral devices. The peripheral devices are coupled to a scan path. A memory storage device that is external to the microcontroller is also coupled to the scan path. When commanded, data is shifted out of each device configuration register onto the scan path and stored in the external memory device. This is particularly useful for obtaining the states of each device without bringing down the application. Furthermore, configuration data stored in external memory can be loaded into the peripheral device configuration registers via the scan path. This invention also supports zero-volt suspend/resume which does not need extra software readable shadow registers which are often required in other architectures for reading back the current state of legacy registers which are read-only.

Brief Summary Text (3):
The apparatus and method according to the invention pertains to peripheral state registers embedded in a microcontrollers, and more specifically, using scan hardware to capture peripheral device states.

Brief Summary Text (9):
The peripheral devices embedded in a microcontroller each have their own individual registers. Typical peripheral device registers include state registers, instruction registers, address registers, status registers and data registers. Depending on the peripheral, certain registers store configuration information needed for the peripheral's proper operation during start up. On system start up, the execution unit initializes each peripheral device with device specific initial configuration data. This initialization could occur during a cold start-up, zero-volt suspend/resume procedure or after a system crash.

Brief Summary Text (10):
In addition to having peripheral devices, many of today's microcontrollers have embedded test circuitry. In 1985, a group of European companies formed Joint European Test Action Group (JETAG) to devise ways to reduce manufacturing costs. One concept was to incorporate such test circuitry into standard components (controlled via software), eliminating the need for sophisticated in-circuit test equipment. This concept gained support in the U.S., where in 1988, several North American companies formed the Joint Test Access Group (JTAG) consortium to formalize the concept. In 1990, the Institute of Electrical and Electronic Engineers (IEEE) refined the concept and created the 1149.1 standard (which is incorporated herein by reference), known as IEEE Standard Test Access Port and Boundary Scan Architecture.

Brief Summary Text (11):
In such architecture, a JTAG test device is connected to a microcontroller and performs a "boundary-scan test" on the microcontroller. Boundary scan cells contain shift register elements that connect together to form a scan chain around the core logic circuit. Input/output (I/O) signals freely pass between integrated circuit

(IC) pins and the core logic, through the boundary scan cells, in normal mode. However, in test mode, only test signals are allowed to pass into or out of the core logic, via a test port and through the boundary scan chain, providing observability and controllability of the input and output signals. The JTAG test commands are typically drawn from a fairly limited set of commands particularly adapted for testing the interconnections of microcontrollers and are not typically well suited for testing or monitoring its internal logic. Instructions and associated data for testing are read serially into each microcontroller peripheral boundary scan cell registers and read out serially, and after the instructions has been carried out the result is read out serially.

Brief Summary Text (12):
While boundary scan techniques are useful in testing interconnection between components, the scan path does not include internal logic registers. In full scan path design, typically all registers and storage elements are connected in the scan paths.

Brief Summary Text (15):
According to the invention, the configuration states of peripheral devices that are embedded in a microcontroller are saved using scan hardware. Various configuration registers, both internal (non-I/O registers) and external (I/O registers), are connected in a configuration scan path internal to the peripheral device. When it is desirable to save the configuration state(s) of the peripheral device(s), the configuration scan path is scanned, and this data can be stored in an external memory device. The external memory device can be either volatile or non-volatile.

Brief Summary Text (16):
In addition, in one embodiment, the scan path is IEEE 1149.1 compliant. Furthermore, according to the invention, various configuration registers, both internal and external, can be saved to an external memory device without intervention of the execution unit of the microcontroller. Instead of the execution unit issuing read and write commands to the various configuration registers and an external memory, configuration data is scanned out to the external memory via a scan path.

Brief Summary Text (17):
Also according to the invention, the configuration states of peripheral devices can be loaded into peripheral device registers using scan hardware. Configuration states can be stored in external memory and shifted into the configuration registers of peripheral devices via a scan path.

Brief Summary Text (18):
Furthermore, according to the invention, configuration states of peripheral devices can be loaded into peripheral device registers that are read-only and saved from peripheral device registers that are write-only by using this scan path.

Drawing Description Text (4):
FIG. 2 is a block diagram illustrating a scan path between peripheral devices;

Detailed Description Text (4):
Scan Hardware

Detailed Description Text (5):
Microcontrollers that are IEEE 1149.1 compliant have boundary scan paths that couple the registers of peripheral device's I/O pins. Boundary scan provides access to the periphery of the microcontroller, but not the internal registers of the execution unit and peripheral devices. In full scan path design, all registers are connected in the scan paths. Scan paths connect individual flip-flops within a register and then connect the registers, e.g., bit one of register one is connected to bit two of register one, and bit two is connected to bit three of register one,

and so on until the last bit of register one is connected to bit one of register two. According to the invention, scan is implemented on sufficient registers to save configuration state of the device. Furthermore, it is not necessary to capture all data from each register to determine the state of the device. For instance, data from a status register may not be needed to determine the configuration state of the device.

Detailed Description Text (6):
FIG. 2 illustrates typical peripheral devices embedded in a microcontroller M with a scan path. An input pin IN of microcontroller M is provided to shift configuration data into each peripheral configuration register. The configuration registers of the clock and power management unit 102, interrupt control unit 106, timer control unit 110, DMA unit 114, PIO unit 132, asynchronous serial port 136, synchronous serial port 140, chip select unit 126 and bus interface unit 118 are daisy chained together via signal line SCAN.sub.-- PATH. For illustrative purposes, the SCAN.sub.-- PATH line from the output of the bus interface unit configuration registers 120a is coupled to the output pin OUT of the microcontroller M. Data is synchronously shifted in or out of each configuration register utilizing clock CLK.sub.-- SCAN. The data out pin OUT is coupled to an external memory 200. Thus, the configuration data from each peripheral device is sequentially shifted out of each configuration register into external memory 200 via SCAN.sub.-- PATH. Likewise, the external memory 200 is coupled to the input pin IN, so that data from external memory 200 can be synchronously shifted into each peripheral configuration register via SCAN.sub.-- PATH.

Detailed Description Text (8):
Peripheral devices embedded in microcontrollers have many different registers that store data for use in a variety of functions. Sometimes, registers necessarily define a configuration state of a peripheral. Other times, registers contain transitory information (such as UART receive status bits) that are not strictly needed to establish the peripheral's configuration. Further, whether a register has information "necessary" to define a peripheral configuration may depend on the particular implementation of the peripheral and its applications. As an illustration, one such peripheral device is an asynchronous serial port 136. A common asynchronous serial port is a Universal Asynchronous Receiver and Transmitter (UART) 300. The UART 300 is one peripheral that provides a serial interface of the microcontroller M. Some of the peripheral's functions include the serialization of parallel data and the insertion of start, parity, and stop bits, or the parallelization of serial data and the separation of start, parity, and stop bits.

Detailed Description Text (11):
FIG. 4 is a block diagram of a scan path coupling the registers of a peripheral device. For illustrative purposes, the registers from the UART 300 are shown. Since it is not necessary to capture the data from all registers, the scan path couples those registers used to define the device's configuration. For example, in this implementation the status registers 312 and 320, receiver shift register 324 and transmitter shift register 326 of the device are left out of the scan path. A clock signal CLK.sub.-- SCAN is provided to the device for synchronous data shifting. The SCAN.sub.-- PATH IN is coupled to the receiver buffer 302. The SCAN.sub.-- PATH IN line could have originated from a SCAN.sub.-- PATH OUT of other peripheral device or from input pin IN. The registers from the receiver buffer 302 are coupled to the data format register 304. Next, the divisor latch (LSB) register 306 is coupled to the divisor latch (MSB) register 308. The divisor latch (MSB) register 308 is then coupled to the modem control register 310. Next, the modem control register 310 is coupled to the interrupt enable register 314, bypassing the modem status register 312. The interrupt enable register 314 is then coupled to the interrupt ID register 316. The interrupt ID register 316 is coupled to the scratch pad register 318. The scratch pad register 318 is coupled to the transmitter hold register 322, bypassing the serialization status register 320. The transmitter hold register 322 is then

coupled to the next peripheral configuration register or output pin OUT.

Detailed Description Text (12):
One skilled in the art could appreciate that inclusion or exclusion of particular registers to be saved can be different from system to system. That is, some predetermined subset of configuration registers could differ according to the system. For example, in a high speed system that quickly restarts, it might be desirable to include status registers in the scan path. Thus, the modem status register 312 and/or serialization status register 320 of the UART 300 can be included in the scan path.

Detailed Description Text (13):
In addition, internal registers (as opposed to I/O registers) can be included in the scan path. The contents of these registers are typically not ascertainable without debug tools or the addition of shadow registers, but it might be desirable to include these registers in the scan path. For example, the receiver shift register 324 and/or transmitter shift register 326 of the UART 300 can be included in the scan path.

Detailed Description Text (17):
Using the configuration scan path according to the invention provides a graceful approach to those problems. Whether registers necessary for peripheral configuration are read-only, write-only, hidden, locked, or otherwise difficult to save and restore using processor reads and writes, the configuration scan chain can capture that information.

Detailed Description Text (18):
Returning to the example of the receiver buffer register 302, because it is coupled to the scan path (as shown in FIG. 4), configuration data can be shifted into this register via SCAN.sub.-- PATH. Likewise, the transmitter hold register 322 typically has write-only capabilities. Data from this register normally cannot be ascertained with I/O commands (or memory commands for memory mapped I/O) from execution unit 124. As shown in FIG. 4, transmitter hold register 322 is coupled to the scan path and configuration data can be shifted out of this register via SCAN.sub.-- PATH.

Detailed Description Text (19):
Finally, an example of a peripheral device with external registers that cannot be accessed easily by software due to the nature of the hardware. One example is the interrupt control unit 106. A common interrupt control unit is the legacy 8259 .ANG. style programmable interrupt controller. This controller uses four initialization control words, namely Initialization Command Words (ICW) ICW1-ICW4, which are written sequentially into the device via a single I/O port to four registers. The four registers can be included as configuration registers 108a and are typically write-only registers. Therefore, the registers cannot be read back out. As shown in FIG. 2, coupling the configuration registers 108a to the scan path, configuration states of the interrupt control unit 106 can be loaded into external memory 200.

Detailed Description Text (22):
IEEE 1149.1 JTAG Boundary Scan and Test Access Port

Detailed Description Text (23):
Another embodiment of the invention utilizes the JTAG boundary scan path and Test Access Port (TAP) hardware to save the configuration data of the peripheral devices or load configuration data into the peripheral devices. For microcontrollers that are IEEE 1149.1 compliant, the scan paths SCAN.sub.-- PATH for each peripheral configuration register can be coupled to the boundary scan. The IEEE 1149.1 standard (May 21, 1990) is described in the publication IEEE Standard Test Access Port and Boundary-Scan Architecture, published by the IEEE, Piscataway, N.J.

(herein incorporated by reference.)

Detailed Description Text (24):
A command instruction is loaded through the TAP. When the registers are clocked, configuration data is shifted out of the peripheral device register onto the boundary scan path. In addition, the memory storage device could be coupled to the Test Access Port for easy load/storage of configuration data.

Detailed Description Text (30):
By saving the configuration states using the scan path, reconfiguration of the peripheral device registers can be accomplished without intervention of the execution unit 124 of microcontroller M. Thus, there would be no need to reload configuration data with I/O commands. As discussed previously, some configuration registers may not be accessible via processor I/O commands because the registers are internal or may be read-only or write-only.

CLAIMS:

1. A microcontroller, comprising: an execution unit; a peripheral device coupled to the execution unit, the peripheral device comprising configuration registers; a means for defining a scan path comprising the configuration registers and for communicating configuration data for the peripheral device; and a means for saving the configuration data via the scan path.

2. A microcontroller, comprising: an execution unit; a peripheral device coupled to the execution unit, the peripheral device comprising configuration registers; a means for defining a scan path comprising the configuration registers and for communicating configuration data for the peripheral device; and a means for loading the configuration data via the scan path.

3. A method of saving configuration data for a peripheral device of a microcontroller, the method comprising the steps of: detecting a command to save configuration data for a peripheral device of a microcontroller; and saving the configuration data via a scan path comprising a configuration register of the peripheral device in response to the command.

8. A method of loading configuration data for a peripheral device of a microcontroller, the method comprising the steps of: detecting a command to load configuration data for a peripheral device of a microcontroller; and loading the configuration data via a scan path comprising a configuration register of the peripheral device in response to the command.

Previous Doc Next Doc Go to Doc#

Refine Search

Search Results -

Terms	Documents
L7 and L2	15

Database: US Pre-Grant Publication Full-Text Database; US Patents Full-Text Database; US OCR Full-Text Database; EPO Abstracts Database; JPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Search: L22

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Thursday, February 16, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L22	L7 and l2	15	L22
L21	L20 and l7	1	L21
L20	L2[ti]	20	L20
L19	L18 and l17	13	L19
L18	L2 same pin	38	L18
L17	L16 and l15	15	L17
L16	scan adj1 command	658	L16
L15	L14 and l7	15	L15
L14	L13 and l2	16	L14
L13	L12 and emulating	16	L13
L12	L11 and scan	23	L12
L11	L2 and jtag	25	L11
L10	L9 same emulating	1	L10
L9	L1 same jtag	102	L9

L8	L7 same l1	2	L8
L7	emulating same scan	140	L7
L6	L2 same emulating	5	L6
L5	L3 and l1	1	L5
L4	L3 same emulating	0	L4
L3	virtual adj1 scan adj1 path	9	L3
L2	embedded adj1 l1	510	L2
L1	microcontroller	38134	L1

END OF SEARCH HISTORY